PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :		(11) International Publication Number:	WO 92/12478
G06F 7/00, 7/22, 11/00 G06F 12/00	A1	(43) International Publication Date:	23 July 1992 (23.07.92)

(21) International Application Number: PCT/US92/00143

(22) International Filing Date: 9 January 1992 (09.01.92)

(30) Priority data:
639,583 9 January 1991 (09.01.91) US

(71) Applicant: VERIFONE, INC. [US/US]; Three Lagoon Drive, Suite 400, Redwood City, CA 94065 (US).

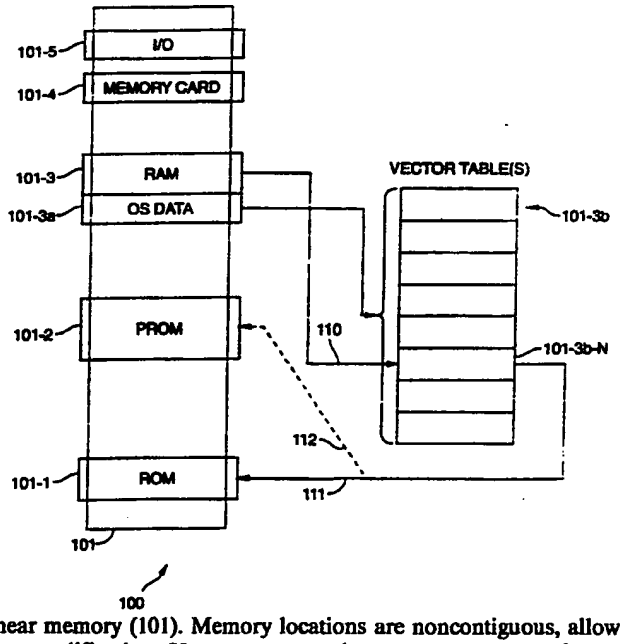
(72) Inventor: ECKLEY, Gordon, Paul, Jr. ; 5925 Happy Pines Drive, Forresthill, CA 95631 (US).

(74) Agents: CASERZA, Steven, F. et al.; Cooley Godward Castro Huddleson & Tatum, Five Palo Alto Square, 4th Floor, Palo Alto, CA 94306 (US).

(81) Designated States: AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH (European patent), CI (OAPI patent), CM (OAPI patent), DE (European patent), DK (European patent), ES (European patent), FI, FR (European patent), GA (OAPI patent), GB (European patent), GN (OAPI patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU (European patent), MC (European patent), MG, ML (OAPI patent), MR (OAPI patent), MW, NL (European patent), NO, PL, RO, RU ,SD, SE (European patent), SN (OAPI patent), TD (OAPI patent), TG (OAPI patent).

Published
With international search report.

(54) Title: TRANSACTION AUTOMATION SYSTEM INCLUDING NOVEL MEMORY ARCHITECTURE AND MANAGEMENT



(57) Abstract

A transaction system (100) includes a linear memory (101). Memory locations are noncontiguous, allowing expansion of any memory area without hardware or software modification. Upon power-up the system accesses software stored in ROM (101-1). This software includes the perating system. Once the system is operating under control of ROM (101-1) programs, vector tables are created to allow the system to locate those portions of memory containing predefined perations such as commands, subroutines, or data files. System operation is turned over to EEPROM (101-2) software which modifies the information in the vect r tables (101-3b). Control is then turned ver to an initialization program in a Battery backed up RAM (101-3), to update the vector tables to define additional files for which there are modified versions in RAM (101-3). Application software in RAM (101-3) then executes, and the system utilizes the vectors for locating the most current version of the software stored in ROM (101-1), EEPROM (101-2), r RAM (101-3).

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	RU	Russian Federation
CG	Congo	KP	Democratic People's Republic of Korea	SD	Sudan
CH	Switzerland			SE	Sweden
CI	Côte d'Ivoire	KR	Republic of Korea	SN	Senegal
CM	Cameroon	LI	Liechtenstein	SU	Soviet Union
CS	Czechoslovakia	LK	Sri Lanka	TD	Chad
DE	Germany	LU	Luxembourg	TG	Togo
DK	Denmark	MC	Monaco	US	United States of America

TRANSACTION AUTOMATION SYSTEM INCLUDING NOVEL MEMORY ARCHITECTURE AND MANAGEMENT

CROSS-REFERENCE TO RELATED APPLICATIONS

The following co-pending applications are related to this application, and are hereby incorporated by reference:

USSN	Filing Date	Title	Attorney Docket Number
639,584	1/9/91	Novel Transaction System Architecture	VERI-125
639,572	1/9/91	Transaction System Including Novel Program Structure	VERI-127
640,279	1/9/91	Method and Structure for Determining Transaction System Hardware & Software Configurations	VERI-129
639,838	1/9/91	Emulator for Use with a Transaction System	VERI-144

Technical Field

This invention pertains to transaction automation devices, such as electronic cash registers (ECRs), point of sale (POS) devices, and the like. More specifically, this invention pertains to a novel transaction automation system including a novel method and structure for providing memory used during the operation of the transaction automation system.

Background

Transaction automation systems of one sort or anoth r are well known in th prior art. Electronic cash registers of varying d grees of sophistication have become rather commonplace in recent years. While the

degree of sophistication of such devices varies, each essentially operates as a computer system performing specific tasks associated with the processing of the desired transactions.

Certain of these prior art ECRs include a microprocessor-based computer system including a factory programmable memory device, typically a mask programmable or one-time programmable (OTP) read-only memory (ROM), which stores sufficient program information to cause the microprocessor to operate in order to control the ECR such that it performs the desired transaction processing function.

In such prior art systems, the factory provided ROM must necessarily define all the functions which the ECR must perform. While this may not seem to pose a problem for simple electronic cash registers, any changes in the function of the cash register will require a physical change of the ROM, which requires technical capabilities beyond that of a cash register operator, and likely beyond that which is available in a given business establishment. Furthermore, there is also a time delay associated with the factory providing the new ROMs, and their installation. In addition, there is the cost associated with the new ROM and its delivery and installation. U.S. Patents 4,688,173 and 4,811,219 both pertain to an electronic cash register including a ROM and one or more additional memories. The additional memories include addresses which are identical to addresses used by the ROM. A technique is used in order to memory map the additional memories onto the address space which serves to address both the ROM and the additional memories. In this manner, a microprocessor capable of addressing a given amount of memory space has access to a larger amount of memory due to the ability of the memory to be bank-switched as needed.

The additional memories are capable of storing modified programs such that the ECR utilizes these modified programs, rather than their unmodified

counterparts contained within the ROM. While th '173 and '219 patents d scribe a structure which allows memory bank switching and is capable of utilizing modified programs, they have the disadvantage of being quite rigid in their hardware requirements, in that once the hardware is designed, changes in memory size necessitate significant hardware changes. Furthermore, program sizes are constrained by the size of memory banks.

SUMMARY OF THE INVENTION

In accordance with the teachings of this invention, a novel transaction automation system and method are taught in which a linear memory structure is used. Various portions of the linear memory structure are mapped to a variety of purposes. In one embodiment, these memory locations are noncontiguous, thereby allowing easy expansion of any of the memory areas without the need for modifying hardware or software in order to access these expanded memory areas. In a preferred embodiment, a variety of memory devices are used for various purposes to distinct advantage.

For example, a read-only memory device is used to store basic information necessary for the microprocessor-based transaction automation system to function when the power is turned on. An additional electrically erasable programmable read-only memory (EEPROM) is used to store additional information, such as modified versions of the programs stored within the ROM, which are not likely to be frequently altered. A battery backed up random access memory (RAM) portion of the memory is used to store programs and data which are more likely to be altered, for example, applications software rather than operating system software. An additional portion of the memory comprises a bulk removable memory d vice, such as a JEIDA memory card. Yet another portion of the memory spac serves as a means for addressing memory mapped I/O devices.

In operation, upon power-up the microprocessor-based transaction automation processing system of this invention accesses the fundamental software stored in the ROM which allows the system to perform basic functions. This fundamental software includes all or part of the operating system and, if desired, a rudimentary portion of the application software. Once the system is operating under the control of the programs stored in ROM, one or more vector tables are created based upon the information stored in ROM so as to allow the system to locate those portions of memory containing predefined operations such as commands, subroutines, or data files. The operation of the system is turned over to the software contained in the EEPROM which then modifies or appends the information stored in the one or more vector tables pertaining to the commands, subroutines, and files provided by the software contained in the EEPROM. In this manner, the EEPROM is able to provide additional functionality by virtue of its control program, and subroutines and commands defined within the EEPROM.

If certain of these commands and subroutines are updated versions of those contained within the ROM, the changes in the vector tables will cause the system to operate upon the modified versions contained in EEPROM as directed by the updated vector tables, rather than the unmodified versions stored within ROM and previously defined in the vector tables.

Once the vector tables have been updated and the tasks defined by the initialization program contained within the EEPROM are completed, control is then turned over to an initialization program contained within the RAM. This initialization program again updates the vector tables in order to define within the vector tables additional commands, subroutines, and files or overwrite the vectors associated with commands and subroutines for which there ar modified versions stored in RAM. Th application software stored in RAM

then executes, and the system utilizes the vectors stored in the one or more vector tables for locating the most current version of the software associated with various portions of the operating system and the applications software as stored in either the ROM, EEPROM, or RAM.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram depicting the memory structure of one embodiment of this invention.

Detailed Description

Figure 1 is a block diagram depicting one embodiment of a memory system of a transaction automation system constructed in accordance with the teachings of this invention. Memory space 101 is capable of being directly accessed by the microprocessor (not shown). Memory space 101 is divided to include a number of memory devices as shown. Preferably, the address location of these memory devices are not contiguous, thereby allowing easy expansion of any given memory device without necessarily requiring modifications to hardware or software in order to access the expanded memory.

Memory device 101-1 serves to store the rudimentary software necessary for the microprocessor-based system to operate when power is turned on. For this reason, memory device 101-1 preferably comprises a nonvolatile memory, such as a mask programmed ROM, or an OTP ROM. Alternatively, although somewhat less desirably, memory device 101-1 may comprise a PROM or a battery-backed up static RAM (SRAM).

Memory device 101-1 includes at least the most fundamental portion of the operating system so that upon power-up or power-on reset the transaction automation system can operate as a fundamental microprocessor computer. In an alt rnative embodiment, certain portions of the operating system are also

included in memory device 101-1 and/or more sophisticated portions of the operating system are also stored in memory device 101-1. Factors which can be used to determine how much of the total operating system and how much, if any, of the application software is to be stored in memory device 101-1, include the amount of space available in memory device 101-1, and the desired amount of functionality upon power-on reset when only the information stored within memory device 101-1 is available to the microprocessor system, for example, due to hardware malfunction.

Upon power-on reset, the microprocessor system boots up and accesses memory device 101-1. An initialization routine stored within memory device 101-1 performs a number of functions. Diagnostics are performed in order to determine whether there are any errors with respect to the software contained within memory device 101-1. Such software diagnostics necessarily also test the portion of the microprocessor system which accesses memory device 101-1, and to a certain extent, the operation of the microprocessor itself.

The initialization routine stored within memory device 101-1 then serves to determine the hardware configuration of the system, for example, the number and types of peripherals such as printers, keyboards, displays, and the like, contained within the transaction automation system. In the event alternative hardware configurations are not permitted or of interest, the hardware configuration operation is not performed.

A hardware diagnostic operation is then performed in order to verify that the hardware is functioning properly. Based upon the hardware devices detected during the hardware configuration step, or a predefined set of hardware where alternative configurations are not permitted, the microprocessor

communicates with each hardware device to determine that it is responding correctly.

Following completion of any one of these steps, or at the completion of all of these steps, an indication is made whether any errors have been detected. Depending on the nature of the error, the system may shut down, as defined by the initialization routine contained within memory device 101-1.

The initialization routine of memory device 101-1 then creates one or more vector tables 101-3b which is stored in the operating system data portion 101-3a of memory device 101-3. Since memory device 101-3 includes a portion for the storage of data, memory device 101-3 is formed of a type of memory which can be easily written and rewritten. In one embodiment, memory device 101-3 comprises a static RAM which is backed up by a battery such that its contents are retained even when the transaction automation system is turned off. Low power CMOS static RAMs are ideal for this purpose and Nicad batteries which are continuously recharged as long as the transaction automation system is turned on provide a convenient backup power source.

Vector tables 101-3b serve to store address locations within memory space 101 where certain commands or subroutines are stored. For example, when an application program indicates that a print command is to be executed in order to cause a slip printer to print an entry associated with the purchase of an item, the operating system will find the address stored in the vector table 101-3b associated with that print command, and in turn address the specific software associated with that print command which is stored at a location within memory space 101 specified by that entry in vector table 101-3b. Since the information stored in memory device 101-1 is fixed, or altered very infrequently, memory device 101-1 may simply include a vector table which is now written into vector table 101-3b. Alternatively, the initialization routine of memory

device 101-1 may locate each command or subroutine stored within memory device 101-1 and create an associated vector within vector table 101-3b.

Following the creation of vector table 101-3b, the initialization program of memory device 101-1 transfers control to an initialization routine of memory device 101-2. Memory device 101-2 is used to store modified programs corresponding to the original programs stored within memory device 101-1. Memory device 101-2 also serves to store more sophisticated portions of the operating system software and, if desired, portions of the applications software. Since memory device 101-2 stores modified versions of software, memory device 101-2 is preferably formed utilizing device types which are capable of being altered. However, since memory device 101-2 does not store volatile data, its contents are typically not altered very frequently, although more frequently than is memory device 101-1, memory device 101-2 may conveniently be formed of one or more EEPROMs. The initialization routine of memory device 101-2 detects the commands and subroutines of the operating system software and applications software stored within memory device 101-2 and updates vector table 101-3b accordingly. During this process, new vectors are added to vector table 101-3b for new portions of the operating system software and applications software stored within memory device 101-2, and old vectors stored within vector table 101-3b are overwritten to redirect memory accesses from prior commands and routines stored in memory device 101-1 to modified versions of those commands and routines stored in memory device 101-2.

Once the initialization routine of memory device 101-2 is completed, control is turned over to the initialization routine of memory device 101-3. Memory device 101-3 stores modified versions of the applications software stored in memory devices 101-1 and 101-2, as well as more sophisticated portions of the applications software. The initialization routine of

memory device 101-3 then modifies vector table 101-3b in order to add additional vectors pertaining to new portions of the applications software stored in memory device 101-3, and updates vector table 101-3b to overwrite vectors pertaining to portions of the applications software corresponding modified versions stored within memory device 101-3. If desired, memory device 101-3 can include additional portions of the operating system, or additional modified portions of the operating system. However, since memory device 101-3 is likely to be the most volatile of the memory devices (although it is backed up by battery) operating system software is preferably not stored in memory device 101-3 although it can be, for example, if memory devices 101-1 and 101-2 are filled to capacity or if trial versions of operating system software are being used.

The initialization routine of memory device 101-3 then turns control over to the applications software which executes in order to perform the transaction automation application. In one embodiment, the application software operates a variety of tasks concurrently by time sharing the capabilities of the microprocessor system. Thus, for example, transparent to the user, tasks pertaining to the electroluminescent displays, printers, keyboard input and numeric calculations are concurrently performed.

As shown in Figure 1, operating system data stored within portion 101-3A of memory device 101-3 includes one or more vector tables 101-3B. When software residing anywhere within memory space 101 is to perform an input/output/library operation, the software access is vector Table 101-3B at an address corresponding to that operation. This is depicted by line 110. Vector Table 101-3 stores at this location the address corresponding to the software to be ex cuted. For example, a portion of th operating system defining a print command is addressed by vector

table 101-3B to point (illustrated by line 111) to a corresponding memory location in memory device 101-1. When the vector table location 101-3B-N is rewritten, for example, when memory device 101-2 includes a modified version of the print sub-routine, the information stored in vector table address 101-3B-N is overwritten such that when a print is to be performed, vector table location 101-3B-N points (as illustrated by dash to line 112) to the corresponding memory location within memory device 101-2.

Memory space 101 also includes, if desired, additional memory devices, for example, a JEIDA memory card 101-4, and memory mapped I/O devices 101-5. Table 1 illustrates one example of a memory map of memory space 101 in accordance with one embodiment of this invention constructed utilizing a Motorola 68302 microprocessor serving as the transaction system CPU. Of interest, memory device 101-3 can easily be expanded utilizing expansion SRAM 101-3 plus as indicated in the memory map of Table 1. By placing the various elements in non-contiguous address locations within memory space 101, such expansion can easily be achieved without the need for software or hardware modifications.

As indicated in Table 1, temporary SRAM area during initial switching serves to transfer control to external RAM; internal dual port RAM and registers serves to support initial diagnostics; and IDMA dummy address for peripheral access serves to support concurrent I/O and computation.

Table 2 depicts one embodiment of a memory map corresponding to the I/O devices which are memory mapped to address locations 101-5 of memory space 101, as depicted in Figure 1.

Table 1

Memory Map

000000 to 3fffff	LOCAL SRAM 101-3	4 Mbyte
400000 to 7fffff	EXPANSION SRAM 101-3+	4 Mbyte
800000 to 803FFF	TEMPORARY SRAM AREA DURING INITIAL SWITCHING	16 Kbyte
804000 to 804FFF	INTERNAL DUAL PORT RAM & REGISTERS	4 Kbyte
805000 to 805fff	IDMA DUMMY ADDRESS FOR ACCESS	4 Kbyte
806000 to 8FFFFF	UNASSIGNED	1 Mbyte minus 14 Kbyte
900000 to 9fffff	I/O DEVICES 101-5	1 Mbyte
a00000 to bfffff	MEMORY CARD 101-4	2 Mbyte

c00000 to dfffff	OPERATING SYSTEM FIXED ROM 101-1	2 Mbyte
e00000 to ffffff	MODIFYABLE ROM 101-2	2 Mbyte

Table 2

I/O MEMORY MAP

	A15	A14	A13	A0	e.g.	
OUTPUT LATCH	0	0	0	1	$900001	
RTC	0	0	1	1	$902001	to $90201F
DUART1	0	1	0	1	$904001	to $90401F
DUART2	0	1	1	1	$906001	to $90601F
MEMORY CARD REGISTERS	1	0	0	1		
SCSI	1	0	1	1	$90A001	to $90A003
UNASSIGNED	1	1	X	1		

All publications and patent applications are herein incorporated by reference to the same extent as if each individual publication or patent application was specifically and individually indicated to be incorporated by reference.

The invention now being fully described, it will be apparent to one of ordinary skill in the art that many changes and modifications can be made thereto without departing from the spirit or scope of the appended claims.

WHAT IS CLAIMED IS:

1. A transaction terminal comprising:
a CPU; and
a linearly addressable memory comprising:

a first memory for storing basic operating system software which is accessed by said CPU upon power-up;

a second memory for storing software selected from the group of additional operating system software, modifications to said basic operating system software, and application software; and

a third memory for storing information selected from the group of additional operating system software, modifications to said operating system software, additional application software, modifications to said application software, and data.

2. A system as in claim 1 wherein said first memory comprises nonvolotile memory.

3. A system as in claim 1 wherein said second memory comprises electrically erasable read only memory.

4. A system as in claim 1 wherein said third memory comprises random access memory.

5. A system as in claim 1 wherein said third memory comprises battery backed up static read only memory.

6. A system as in claim 1 which further comprises means for addressing additional devices as memory mapped I/O devices.

7. A system as in claim 1 which further comprises a bulk removable memory device.

8. A system as in claim 1 which further comprises one or more vector tables for storing pointers to memory addressed of the most current version of selected portions of said operating system software, application software, and/or data.

9. A system as in claim 1 wherein:

said first memory further comprises a first set of instructions for execution by said CPU to initialize said system to utilize the information stored in said first memory and to construct one or more vector tables containing pointers to address locations within said first memory;

said second memory further comprises a second set of instructions for execution by said CPU as instructed by said first set of instructions, to utilize the information stored in said second memory and to update said one or more vector tables by appending additional pointers to address locations within said second memory and/or modifying pointers to address locations when information contained within said second memory is to replace or modify information within said first memory.

10. A system as in claim 9 wherein said third memory further comprises a third set of instructions for execution by said CPU as instructed by said second set of instructions, to utilize the information stored in said third memory and to update said one or more vector tables by appending additional pointers to address locations within said third memory and/or modifying pointers to address locations when information contained within said third memory is to replace or modify information within said first or second memories.

11. A system as in claim 10 which further comprises a fourth memory, wherein said fourth memory furth r comprises a fourht set of instructions for execution by said CPU as instructed by said third set of instructions, to utilize the information stored in said fourth memory and to update said one or more vector tables by appending additional pointers to address locations within said fourth memory and/or modifying pointers to address locations when information contained within said fourth memory is to replace or modify information within said first, second, or third memories.

12. A system as in claim 11 wherein said first memory comprises nonvolatile memory.

13. A system as in claim 12 wherein said second memory comprises electrically erasable, programmable read only memory.

14. A system as in claim 13 wherein said third memory comprises random access memory.

15. A system as in claim 14 wherein said third memory comprises battery backup random access memory.

16. A system as in claim 15 wherein said bulk removal memory comprises a memory cartridge or a memory card.

17. A system as in claim 1 wherein information stored in one or more said memories are stored in noncontiguous locations.

18. A system as in claim 9 wherein said vectors are st red in one of said memory devices.

19. A system as in claim 18 wherein said memory which stores said vectors comprises volatile memory.

1/1

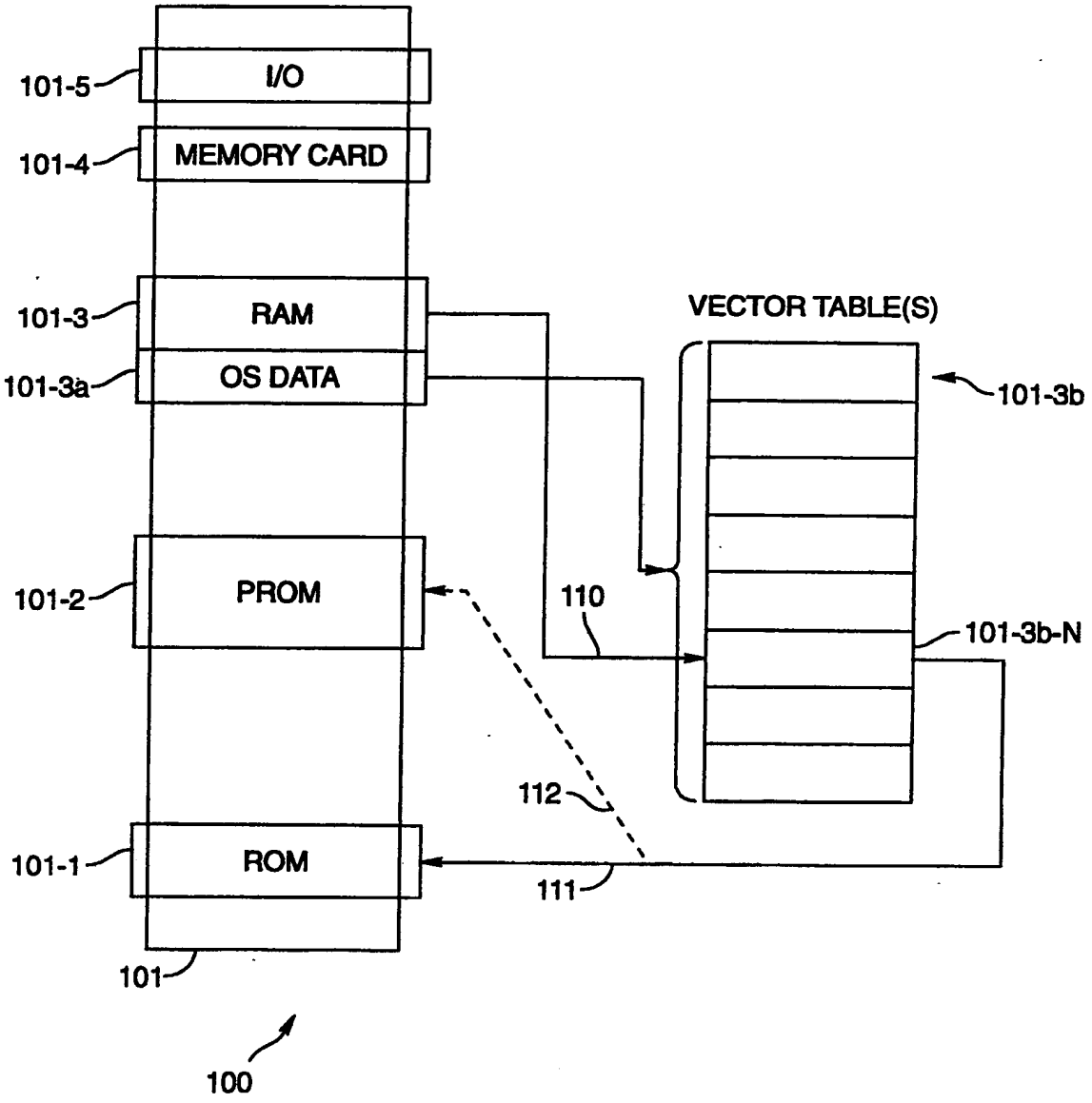

I/O
MEMORY CARD
RAM
OS DATA
PROM
ROM
VECTOR TABLE(S)
101-5
101-4
101-3
101-3a
101-2
101-1
101
100
101-3b
101-3b-N
110
112
111

FIGURE 1

INTERNATIONAL SEARCH REPORT

International Application No. PCT/US92/00143

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [6]

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC (5): G06F 7/00, 7/22, 11/00, 12/00
U.S.Cl.: 395/700; 371/55

II FIELDS SEARCHED

Minimum Documentation Searched [7]

Classification System	Classification Symbols
U.S.	395/700; 371/55

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [9]

Category *	Citation of Document, [11] with indication, where appropriate, of the relevant passages [12]	Relevant to Claim No. [13]
X Y	US, A, 4,811,219 (TOUJI ET AL) 07 March 1989 See the entire document	1-4,6,8-14,17-19 5,7,15,16
X	US, A, 4,688,173 (MITARAI ET AL.) 18 August 1987 See the entire document.	1-19
Y	US, A, 4,980,856 (UENO) 25 December 1990 See the entire document.	16

* Special categories of cited documents: [10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
26 February 1992	14 APR 1992
International Searching Authority	**Signature of Authorized Officer**
ISA/US	John Q. Chavis